M-SEC-905

Offensive Security and exploitation Pentest Report

Marian Bret

Ka Po Chau

Nejma Belkhanfar

Table of Contents

M-SEC-905 0

Table of Contents 1

Methodology 2

Scanning 3

Vulnerabilities 4

Methodology

Work methodology :

We tried to keep the same methodology for every single information gathering or exploitations we did on the scope 10.10.10.0/24.

The recommended methodology about pentesting is this pattern : Footprinting > Network scanning > Enumeration > Exploitation.

First part of the work was to scan the network scope to find the first useful information : the hosts that were up and ports that were open on those hosts.

As we spotted some DNS open ports, we decided to scan them to gather as much information as possible (logs available in 'log-10.10.10.10-53' and 'log-10.10.10.11-53'). Those scans allowed us to get some useful information about hosts and PTR (which are "reverse DNS" that give the domain name associated with an IP). Those information was crucial for deeper scans and possible exploitations.

After getting the whole list of connected hosts and open ports that belong to those hosts, we continued to follow the methodology of pentesting which led to some deeper network scans on a specific host and port (the list is available on 'scan-logs'). We usually used Kali Linux's built-in scanners softwares such as Nmap (https://nmap.org/) or Dirb for http ports as an example. Those scanners have multiple parameters and flags that allow them to do a specific and precise scan to gather information such as services that are running currently on the network with their specific versions, actual configurations, responses of the server…

During those scans, we used some built-in scripts in scanning softwares such as nmap-script-engine (https://nmap.org/man/fr/man-nse.html) or Metasploit scanners that can led to leak of some impacting informations on the victim host (either with the -sC Nmap flag that launch the default script for a port or either precise scripts, using the —script= flag on Nmap).

This scanning work gave us a good overview of the network scope and possible entries vectors to look for possible vulnerabilities deeper.

Keeping on following the pentest methodology, we tried to enumerate as much information as possible on a host/port we found open earlier. To achieve this, we used some softwares and databases such as searchsploit, exploit-db, CVE-details and built-in enumeration scripts (such as SMB enumeration or enum4linux for example).

With all those scanning and enumerating information, we can try to find the good attack vector on a specific host. At this point, we got two options, either the host/port is vulnerable and we managed to exploit it to find some very impacting information or established a root connexion (or vertical/horizontal privilege escalation) or either we found some vulnerabilities and security flaws that haven't been exploited but can led to future exploits. As long as we didn't fulfil in breaking the protections, we continued to follow the methodology in a way to do deeper scans or enumerate more useful information about the system. In fact, we understood that a machine or service we tried to break that led to nothing doesn't mean that the service isn't vulnerable or dangerous for the system (as an example, a service can leak no information but can be vulnerable for Dos attack or else).

The structure we used to store the information we gathered and our exploits is available on the project Github (https://github.com/marianbret/offensive_security_tek5) with this actual PDF report and some logs that follow the same pattern : log-host-port.txt (for example: log-10.10.10.10-22.txt). Those logs contain the scanning information, the enumerations and exploitations.

Scanning

Scanning the project scope:

The project scope is : 10.10.10.0/24

The first machine of the scope is : 10.10.10.1

The last machine of the scope is : 10.10.10.254

The scope scanning logs are available on Github in the 'scans-logs.txt' file.

Scanning SSH ports:

The SSH ports scanning are available on Github in the 'scans-ssh-logs.txt' file.

During those scans, we spotted that some host-keys are duplicated ones on different machines, that could be a huge risk as an attacker which would have compromised one of those machines could easily use shared ssh keys found in the victim machine to attack other machines with the same host-keys in MITM attacks for example.

Scanning DNS ports :

The DNS ports scanning are available on Github in the 'scans-logs.txt' file in the DNS scans section.

Vulnerabilities

10.10.10.22:139:445 :

During the scanning process, we found open ports 139 and 445 on the machine 10.10.10.22. Those ports are SMB ports which is an ‘interprocess-communication protocol’ on Windows. Here it uses Samba which is a software that implements SMB protocol on Unix systems.

We scanned those ports deeper and found that SMB v1 , v2 and v3 were available on the server which can lead to some impacting vulnerabilities that can be related to one of each versions.
Nmap scans are available on ‘log-10.10.10.22-139-445’ and gave us some useful information. This version is vulnerable for regsvc-dos exploit which leads to a possible Ddos attack that will crash the system.
Nmap smb-enum-shares script, enum4linux and smbmap softwares gave us a lot of information such as the Anonymous access rights on the Public directory, the users on the system (myles and guest), the OS information and folders tree.
We succeeded in connecting to the Public shared folder (all exploitations are available in the log linked above) and found multiple files. One of those was a password protected zip file, we used the “get” command to fetch all the files from the smb shared drive to our local machine.
With the help of the zip2John software, we created a hash file from the zip archive and let JohnTheRipper crack the password with the built-in rockyou wordlist file (fcrackzip software works perfectly too).
The zip file was cracked, while trying to run the pmanager file, we found that we need some information to fulfill file’s parameters that were an username (myles, we had the information by enum4linux) and a UserId (that was in plain text in myles’s card picture) we had those data in the enumeration part of the work on this port. The pmanager script returned a password for myles user that we used to connect to myles’s smb profile successfully. We found some nice information on this smbclient, a todo list which talks about vulnerabilities on thermostat machines and a ssh private key which is encrypted with Ccrypt software (we didn’t try any bruteforce on this encryption).

10.10.10.53:21 :

During the scanning process, we found open port 21 on the machine 10.10.10.53. The port 21 is a ‘FTP’ port which is a ‘file transfer protocol’.

We scanned this port deeper and found out that this actual ftp service allows Anonymous login (nmap built-in default script for ftp port) and the version of the service is vsftpd 2.3.4.
We managed to connect to the host as an anonymous user without any passwords and had the READ right on it. The server as anonymous was empty, nevertheless, we advise the administrator to connect to Cpanel and uncheck “Anonymous access” and “Anonymous upload” boxes, in a way to avoid this kind of connection.

In the information gathering process, we found with the help of searchsploit tool (which is an exploit database researcher tool) that this precise version of vsftpd is vulnerable with a backdoor (CVE-73573) attack which led to a root shell on the system (built-in msfconsole or in exploit-db as a python script). We runned the exploit and managed to get the root access (all scans and exploitations are available on ‘log-10.10.10.53-21’), we dumped the /etc/passwd and /etc/shadow files to crack the passwords with

JohnTheRipper tool and managed to get a credential which is fern11:naruto1. We got some ssh configuration and ssh keys too (public and private). It was accessible through the backdoor exploit root shell, on ftp connecting as fern11 or while connecting as fern11 user on ssh 22 port on the same machine.